Principal Component Analysis

1 Introduction

Let $p(x)$ be a multivariate probability density function. For PCA this is typically the normal density function $N(\mu, \Sigma)$, where μ, Σ are the sample mean and variance respectively.

$$\mu = E[x]$$

$$\Sigma = E[(x - \mu)(x - \mu)^T]$$

Given Data points $x_1, x_2 \dots x_n$, from this distribution, the sample mean and the sample covariance are defined as,

$$\hat{\mu} = \frac{1}{n}\Sigma_{i=1}^{n} x_i$$

$$\hat{\Sigma} = \frac{1}{n-1}\Sigma_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T$$

The principal components are the eigen vectors of the sample covariance matrix. PCA projects the vectors $x_1 \dots x_n$ onto the principal components.

The Eigen Vectors of Σ are the left singular vectors of $\frac{1}{\sqrt{n-1}}[(x_1 - \mu)(x_2 - \mu)^T]$.

Thus PCA is just the SVD of *mean centered data* . However the PCA and the SVD can be different for a given set of data as shown in figure 1.

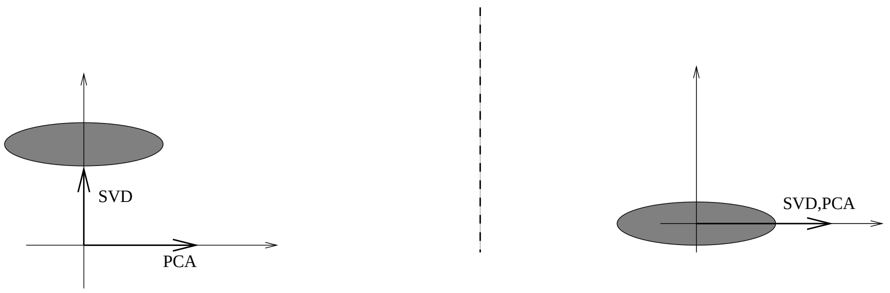

Figure 1:

2 Clustering

Clustering could be defined as partitioning a given set of objects such that the objects placed in each of the clusters produced are *similar*.

Usually ,the Clustering problem is defined as an optimization problem in which a given objective function needs to be optimized. A good objective function is critical for coming up with a good solution. The objective function used in Clustering must match the users subjective criterion. Coming up with a good clustering objective function requires an understanding of the nature of the different functions used and also the type of clusters produced by each of these. Usually, invariances of the data set help in choosing an objective function.

For example, the sum of squares criterion is invariant to translations and rotations, however the *Manhattan metric* as shown in figure 2 is not.

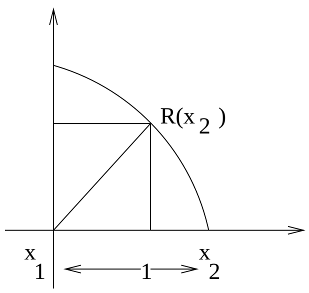

Figure 2: The manhattan metric is not invariant to rotations.

Another decision that one must make, is whether the input data can be scaled or whether it can be normalized.

While trying to locate an extremum of the objective function, there is a tradeoff between time and accuracy as most clustering objectives, are NP-Hard. For example the sum-of-squares is NP-Hard even for $k = 2$.

However ,clustering is a very important problem. It helps in understanding the intrinsic structure of the set. Inspite of all these difficulties, we need to do cluster analysis . Another important problem is to evaluate the clusters output by an algorithm. One way to do this is to, visualize them. However, data sets are in very high dimensions and it is not obvious how they can be visualized. This problem will be studied in a later lecture.

2.1 Sample Clustering algorithms

Hierarchical Clustering , involves constructing a tree to capture the clustering of objects. It can either be *agglomerative* (bottom up) or *divisive*(top down).

The basic idea in *Hierarchical Agglomerative Clustering* is to start out with n-clusters, i.e. each sample in a separate cluster and at every step merge two clusters, till the required number of clusters are obtained.

Pseudocode: Let $\hat{c} = n$ and $C_i = \{x_i\}, i = 1 \dots n$.

> While $\hat{c} \leq k$,
> > Find "nearest" pair of distinct clusters C_i and C_j.
> > Merge C_i and C_j, delete C_j.
> > $\hat{c} \leftarrow \hat{c - 1}$.
>
> end while.

Different definitions of the distance between clusters leads to different clustering algorithms. Examples of distances between clusters are,

$$d_{min}(C_i, C_j) = min_{x \in C_i, x' \in C_j} ||x - x'||$$

$$d_{max}(C_i, C_j) = max_{x \in C_i, x' \in C_j} ||x - x'||$$

$$d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \Sigma_{x \in C_i} \Sigma_{x' \in C_j} ||x - x'||$$

$$d_{mean}(C_i, C_j) = ||m_i - m_j||$$

The clusters produced by these different metrics can differ a lot. Using the d_{min} metric tends to favor elongated clusters, it is also referred to as a *single-linkage* algorithm. For $k = 1$, this is equivalent to *Kruskal's* minimum spanning tree algorithm. The complexity of this algorithm is $O(n^2)$.

Use of d_{max} leads to the farthest neighbor algorithm. In this the formation of elongated clusters is discouraged. If the diameter of a cluster is defined as largest distance between points in the cluster, d_{max} is the diameter of their union. The complexity of this algorithm is $O(n^2)$.